CLEAN CLAIMS AFTER *RESPONSE A*

1. (First Amended per A) A computer implemented method for emulating execution of legacy instructions, where said legacy instructions have instruction addresses, comprising:

accessing blocks of said legacy instructions, said blocks having block addresses,

storing translations, into a translation store, for each of the legacy instructions,

storing translation indications, for indicating translated blocks, into an indexing table at block numbers determined by said block addresses, said storing translation indications using a subset of block address digits whereby block numbers in said table are the same for multiple different blocks,

executing said translated instructions to emulate said legacy instructions,

where for each of the legacy instructions of a translated block having a block number in said table, said storing translations step includes translating the legacy instructions into one or more translated instructions for emulating the legacy instructions, and

if the legacy instruction is not a store instruction, going to said step of executing said translated instructions,

if the legacy instruction is a store instruction, where the store instruction stores to a particular block with a particular block number in said table, checking the indications in said table for the particular block number and,

if the indications indicate that said particular block has not been translated, going to said step of executing said translated instructions,

if the indications indicate that said particular block has been translated, checking said translation store to determine if legacy instruction data has been modified and if modified, repeating the step of translating the legacy instructions and going to said step of executing said translated instructions; and otherwise, if legacy instruction data has not been modified, going to said step of executing said translated instructions.

2. (Original) The method of Claim 1 wherein said step of storing translation indications stores indications for only a subset of all the translated blocks.

3. (Original) The method of Claim 2 wherein said subset of all the translated blocks is stored in a cache.

4. (Cancelled).

5. (First Amended per A) The method of Claim 1 wherein said block address digits are included in a three digit hexadecimal address field and said subset of block address digits is the center digit.

6. (Original) The method of Claim 1 wherein said legacy instructions are for a legacy system having a S/390 architecture.

7. (Original) The method of Claim 1 wherein said legacy instructions are object code instructions compiled/assembled for a legacy architecture.

8. (Original) The method of Claim 1 wherein said legacy instructions include store instructions for modifying instruction code.

9. (Original) The method of Claim 1 wherein said translation indications include a state field for each block number indicating whether the block represented by said block number has been modified.

10. (First Amended per A) The method of Claim 1 wherein,

said subset of all the translated blocks is stored in a cache,

said translation indications include a state field storing a count for each block number indicating whether the block represented by said block number has been modified,

said count in a state field is incremented each time a block represented by said block number has been modified in said cache,

said count in a state field is decremented each time a block represented by said block number has been removed from said cache,

said step of checking said translation store occurs only when said count is zero.

11. (First Amended per A) A computer implemented method for dynamic emulation of object code legacy instructions, where the legacy instructions have instruction addresses determined by compilation/assembly of source code and where the legacy instructions include self-modifying store instructions for modifying instruction code, comprising:

accessing blocks of said legacy instructions, said blocks having block addresses,

storing translations, into a translation store, for each of the legacy instructions,

storing translation indications, for only a subset of all the translated blocks, into an indexing table at block numbers determined by said block addresses, said storing translation indications,

using a subset of block address digits whereby block numbers in said table are the same for multiple different blocks,

including a state field storing a count for each block number indicating whether the block represented by said block number has been modified by self-modifying store instructions,

executing said translated instructions to emulate said legacy instructions,

where for each of the legacy instructions of said subset of all the translated blocks having a block number in said table, said storing translations step includes translating the legacy instruction into one or more translated instructions for emulating the legacy instruction and storing said translated instructions in a cache,

if the legacy instruction is not a store instruction, going to said step of executing said translated instructions,

if the legacy instruction is a store instruction, where the store instruction stores to a particular block with a particular block number in said table, checking the indications in said table for said particular block number and,

if the indications indicate that said particular block has not been translated, going to said step of executing said translated instructions,

if the indications indicate that said particular block has been translated, checking said translation store to determine if legacy instruction data has been modified and if modified, repeating the step of translating the legacy instructions and going to said step of executing said translated instructions; and otherwise, if instruction data has not been modified going to said step of executing said translated instructions.

12. (First Amended per A) The method of Claim 11 wherein said count in a state field is incremented each time a block represented by said block number has been modified in said cache, said count in a state field is decremented each time a block represented by said block number has been removed from said cache, said step of checking said translation store occurs only when said count is zero.

13. (Original) The method of Claim 11 wherein said legacy code is compiled/assembled for a native architecture and executes as a guest on a host architecture.

14. (Original) The method of Claim 13 wherein the native architecture employs CISC instructions and the host architecture employs RISC instructions.

15. (First Amended per A) A computer system for emulating execution of legacy instructions, where said legacy instructions have instruction addresses, comprising:

a group access unit for accessing blocks of said legacy instructions, said blocks having block addresses,

a translator for translating the legacy instructions to form translated instructions,

a translation store for storing the translated instructions,

an execution unit for executing said translated instructions to emulate said legacy instructions,

an index table for storing translation indications for indicating translated blocks at block numbers determined by said block addresses, said index table storing translation indications using a subset of block address digits whereby block numbers in said table are the same for multiple different blocks,

where for each of the legacy instruction of a translated block having a block number in said table, said translation store includes one or more translated instructions for emulating the legacy instruction, and,

if the legacy instruction is not a store instruction, the computer system goes to the execution unit for executing said translated instructions,

if the legacy instruction is a store instruction, where the store instruction stores to a particular block with a particular block number in said table, the computer system checks the indications in said table for said particular block number and, if the indications indicate that said particular block has not been translated, the computer system goes to the execution unit for executing said translated instructions,

if the indications indicate that said particular block has not been translated, said translation store is checked to determine if instruction data has been modified and, if modified, the translator repeats translating the legacy instructions and the computer system goes to the execution unit for executing said translated instructions, and otherwise, if instruction data has not been modified, the computer system goes to the execution unit for executing said translated instructions.

16. (Original) The system of Claim 15 wherein said index table stores indications for only a subset of all the translated blocks.

17. (Original) The system of Claim 16 including a cache and wherein said subset of all the translated blocks is stored in said cache.

18. (Cancelled).

19. (First Amended per A) The system of Claim 15 wherein said block address digits are included in a three digit hexadecimal address field and said subset of block address digits is the center digit.

20. (Original) The system of Claim 15 wherein said legacy instructions are for a legacy system having a S/390 architecture.

21. (Original) The system of Claim 15 wherein said legacy instructions are object code instructions compiled/assembled for a legacy architecture.

22. (Original) The system of Claim 15 wherein said legacy instructions include store instructions for modifying instruction code.

23. (Original) The system of Claim 15 wherein said index table includes a state field for each block number indicating whether the block represented by said block number has been modified.

24. (First Amended per A) The system of Claim 15 wherein,

said system includes a cache for storing said subset of all the translated blocks,

said index table includes a state field storing a count for each block number indicating whether the block represented by said block number has been modified,

said count in a state field is incremented each time a block represented by said block number has been modified in said cache,

said count in a state field is decremented each time a block represented by said block number has been removed from said cache,

said translation store is not checked only when said count is zero.

CLEAN CLAIMS AFTER *RESPONSE A*

25. (First Amended per A) A computer system for dynamic emulation of object code legacy instructions, where the legacy instructions have instruction addresses determined by compilation/assembly of source code and where the legacy instructions include self-modifying store instructions for modifying instruction code, comprising:

a group access unit for accessing blocks of said legacy instructions, said blocks having block addresses,

a translation store for storing translation information for each of the legacy instructions,

an index table for storing translation indications, for only a subset of all the translated blocks at block numbers determined by said block addresses, said index table storing translation indications using a subset of block address digits whereby block numbers in said table are the same for multiple different blocks and including a state field storing a count for each block number indicating whether the block represented by said block number has been modified by self-modifying store instructions,

a cache for storing translated instructions,

an execution unit for executing said translated instructions to emulate said legacy instructions,

a legacy code translator operating, for each of the legacy instructions of said subset of all the translated blocks having a block number in said table, to translate the legacy instructions into one or more translated instructions for emulating the legacy instruction and to store said translated instructions in the cache and,

if the legacy instruction is not a store instruction, the computer system goes to said execution unit for executing said translated instructions,

if the legacy instruction is a store instruction, where the store instruction stores to a particular block with a particular block number in said table, the computer system checks the indications in said table for said particular block number and,

if the indications indicate that said particular block has not been translated, the computer system goes to said execution unit for executing said translated instructions,

if the indications indicate that said particular block has been translated, the computer system checks to determine if instruction data has been modified and if modified, the computer system goes to said translator to translate the legacy instructions into one or more translated instructions and the computer system goes to said execution unit for executing said translated instructions; and otherwise, if instruction data has not been modified, the computer system goes to said execution unit for executing said translated instructions.

26. (First Amended per A) The system of Claim 25 wherein said count in a state field is incremented each time a block represented by said block number has been modified in said cache, said count in a state field is decremented each time a block represented by said block number has been removed from said cache, said bypassing said repeating occurring only when said count is zero.

27. (Original) The system of Claim 25 wherein said legacy code is compiled/assembled for a native architecture and executes as a guest on a host architecture.

28. (Original) The system of Claim 27 wherein the native architecture employs CISC instructions and the host architecture employs RISC instructions.